BigData environment

at PoliCloud

Interdepartmental Research Laboratory

Mario Marchente

DEIB - Politecnico di Milano

23/09/2014

BigData

Nowadays data world :

- is structured and (much more) unstructured
- is generated from many new sources (interactive applications, websites, sensors, meters, and other data-generating machines).
- is characterized by great volumes (tera/peta bytes)
- benefits from the enormous decrease of storage, processing, and bandwidth costs and from the increasing network access.

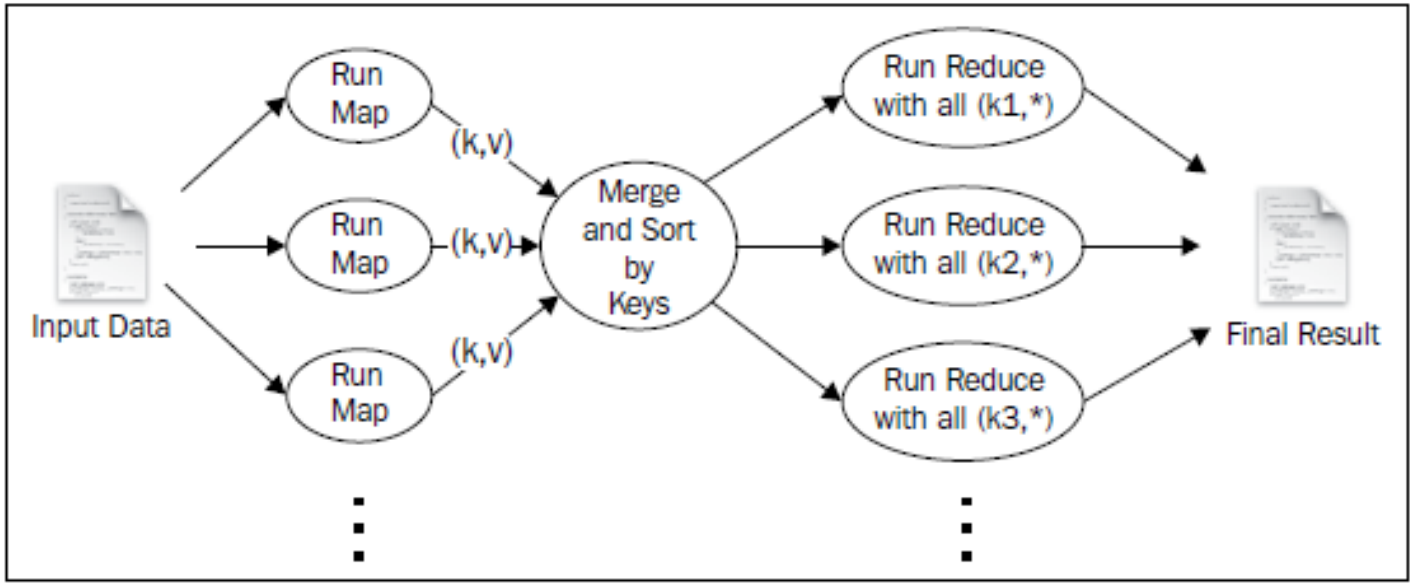
"Big Data is now almost universally understood to refer to the realization of greater business intelligence working with data that was previously ignored due to the limitations of traditional data management technologies." (Harness the Power of Big Data – 2012, McGrawHill)

What is BigData

New data management and analytic technologies are being implemented to complement rather than replace traditional approaches to data management and analytics.

The MapReduce paradigm

MapReduce is a framework for executing highly parallelizable and distributable algorithms across huge data sets using a large number of commodity computers. Google built it and called this computing paradigm MapReduce, based on the following "map" and "reduce" functional programming paradigm:



The MapReduce paradigm

- **Map** In contrast with traditional relational database–oriented information— which organizes data into fairly rigid rows and columns that are stored in tables — MapReduce input data into the map() function *as key/value pairs*. The map() function then produces one or more intermediate values along with an output key from the input.

- **Reduce** - After the Map phase is over, all the intermediate values for a given output key are combined together into a list. The reduce() function then combines the intermediate values into one or more final values for the same key.

The MapReduce paradigm

In a typical MapReduce environment:

- **Components will fail at a high rate:** normally used inexpensive, commodity hardware.
- **Data will be contained in a relatively small number of big files:** each file will be 100 MB to several GB.
- **Data files are write-once:** but they can be appended.
- **Lots of streaming reads:** many threads accessing files at any given time.
- **Higher sustained throughput across large amounts of data:** MapReduce implementations work best when process consistently and predictably colossal amounts of information across the entire environment,as opposed to achieving irregular sub-second response on random instances.

Apache Hadoop

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

Other Hadoop-related projects at Apache include:

- **Ambari™**: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop. Ambari also provides a dashboard for viewing cluster health such as heatmaps and ability to view MapReduce, Pig and Hive applications visually alongwith features to diagnose their performance characteristics in a user-friendly manner.
- **Avro™**: A data serialization system.
- **Cassandra™**: A scalable multi-master database with no single points of failure.
- **Chukwa™**: A data collection system for managing large distributed systems.
- **HBase™**: A scalable, distributed database that supports structured data storage for large tables.
- **Hive™**: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **Mahout™**: A Scalable machine learning and data mining library.
- **Pig™**: A high-level data-flow language and execution framework for parallel computation.
- **Spark™**: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.
- **Tez™**: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases. Tez is being adopted by Hive™, Pig™ and other frameworks in the Hadoop ecosystem, and also by other commercial software (e.g. ETL tools), to replace Hadoop™ MapReduce as the underlying execution engine.
- **ZooKeeper™**: A high-performance coordination service for distributed applications.

Apache Hadoop

A project to develop open-source software library framework for reliable, scalable, distributed computing, that:

- allows for the distributed processing of large data sets across clusters of computers using simple programming models.
- is designed to scale up from single servers to thousands of machines, each offering local computation and storage.
- delivers a highly-available service detecting and handling failures at the application layer.

Hadoop cluster components are:

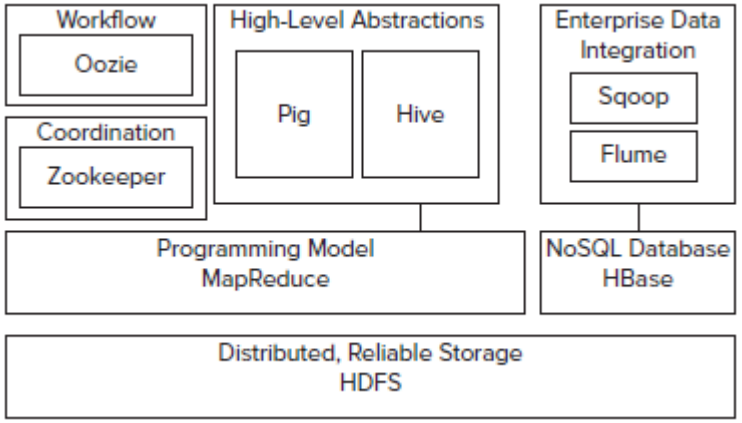
- **Master node** (few) with the following processes:
  - **JobTracker:** interacts with client applications. It is also responsible for distributing MapReduce tasks to particular nodes within a cluster.
  - **TaskTracker:** is capable, in the cluster, of receiving tasks (including Map, Reduce, and Shuffle) from a JobTracker

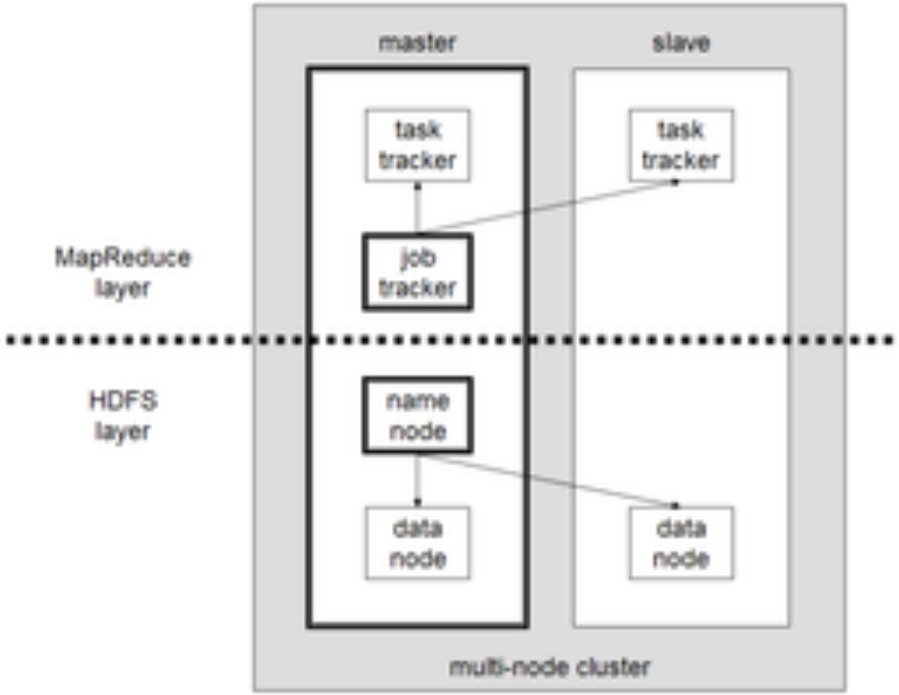
Apache Hadoop

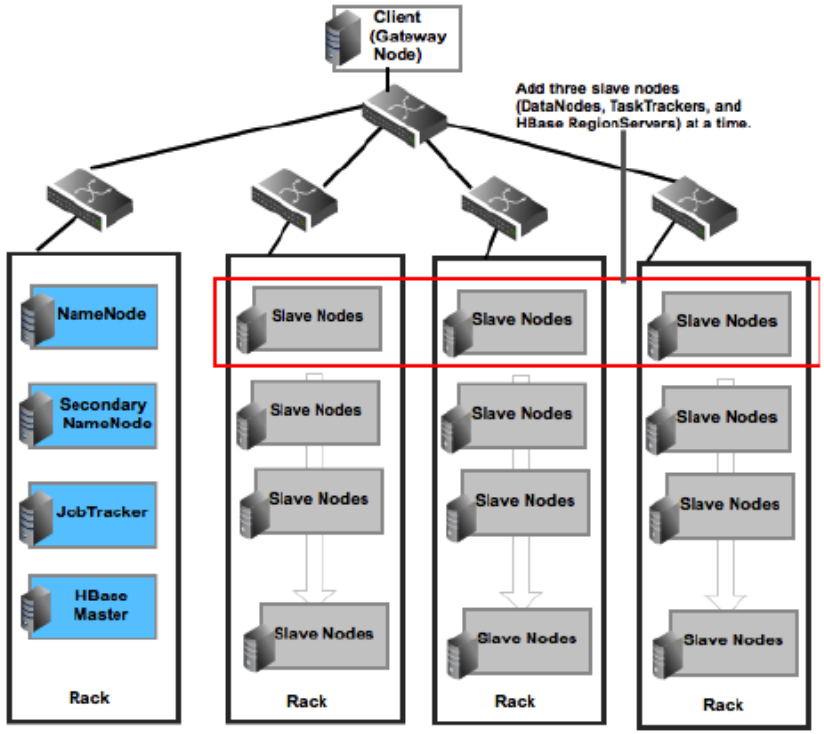
- **NameNode**: : a) stores a directory tree of all files in the HDFS), and b) keeps track of where the file data is within the cluster. Client applications contact NameNodes when they need to locate a file, or add, copy, or delete a file.
- **Data Nodes** a) stores data in the HDFS, and b) is responsible for replicating data across clusters. Interact with client applications, when the NameNode has supplied the DataNode's address.

- **Worker nodes** (many) which provide enough processing power to analyze a few hundred terabytes up to one petabyte. Each worker node includes a DataNode or a TaskTracker, or both.

Apache Hadoop





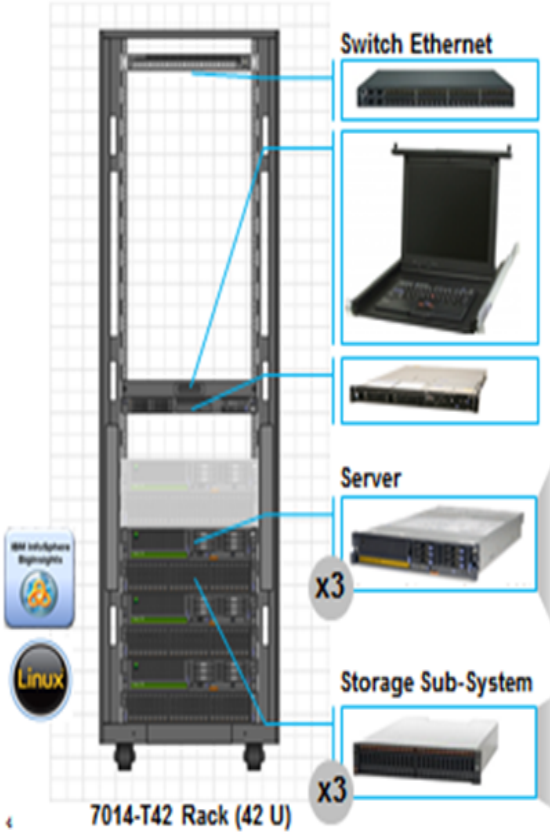


BigData HW Configuration

IBM BigData Lab @ PoliMI

IBM

Lab Infrastructure: Physical lay-out (BigInsight Systems)



1455-24E Switch Ethernet - 24 port 10 Gb/s
- Active ports:
  - 8 sfp 10 Gb/s SX
  - 8 sfp 1 Gb/s TX

7316-TF3 Flat Panel Console

7042-CR7 Hardware Management Console

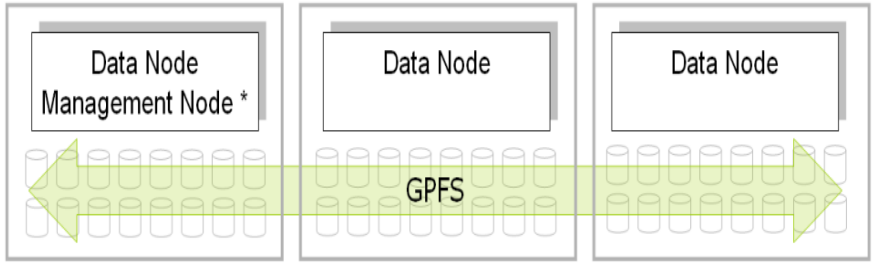
8246-L2T IBM Power Linux 7R2
- 16 core POWER7+ 4.2 Ghz
- 128 GB RAM
- 1 Ethernet 4 porte 1 Gb/s Tx (RJ45)
- 1 Ethernet 2 porte 10 Gb/s Sx (Fibre)
- 6 disks 600 GB 10k rpm
- RedHat EL 6
- IBM POWERVM per Linux
- IBM GPFS

6U

EL1S EXP24S SAS SFF Gen 2
- 16 SAS disks 600 GB/s 10k rpm

6U

A 3-nodes Cluster for Infosphere BigInsight on PowerLinux



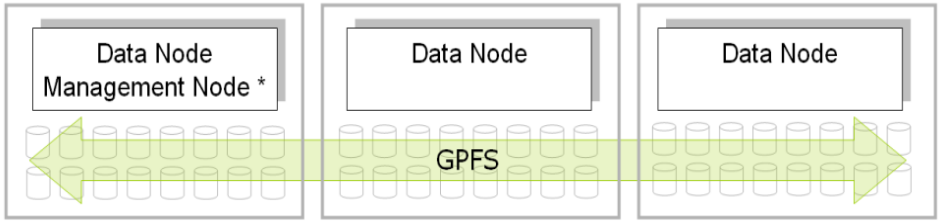
Sistema IBM Power Linux 7R2
- 16 core POWER7+ 4.2 Ghz / 128 GB RAM
- Linux RedHat EL 6

Storage System Expansion

* The following management functions need to be hosted
- ✓ Hadoop JobTracker
- ✓ Hadoop NameNode
- ✓ Hadoop Secondary NameNode
- ✓ BigInsights Console
- ✓ NameNodes not necessary if using GPFS file system in place of Hadoop HDFS file system

BigData HW Configuration

A 3-nodes Cluster for Infosphere BigInsight on PowerLinux



Sistema IBM Power Linux 7R2

- 16 core POWER7+ 4.2 Ghz / 128 GB RAM
- Linux RedHat EL 6

Storage System Expansion

* The following management functions need to be hosted

- ✓ Hadoop JobTracker
- ✓ Hadoop NameNode
- ✓ Hadoop Secondary NameNode
- ✓ BigInsights Console
- ✓ NameNodes not necessary if using GPFS file system in place of Hadoop HDFS file system

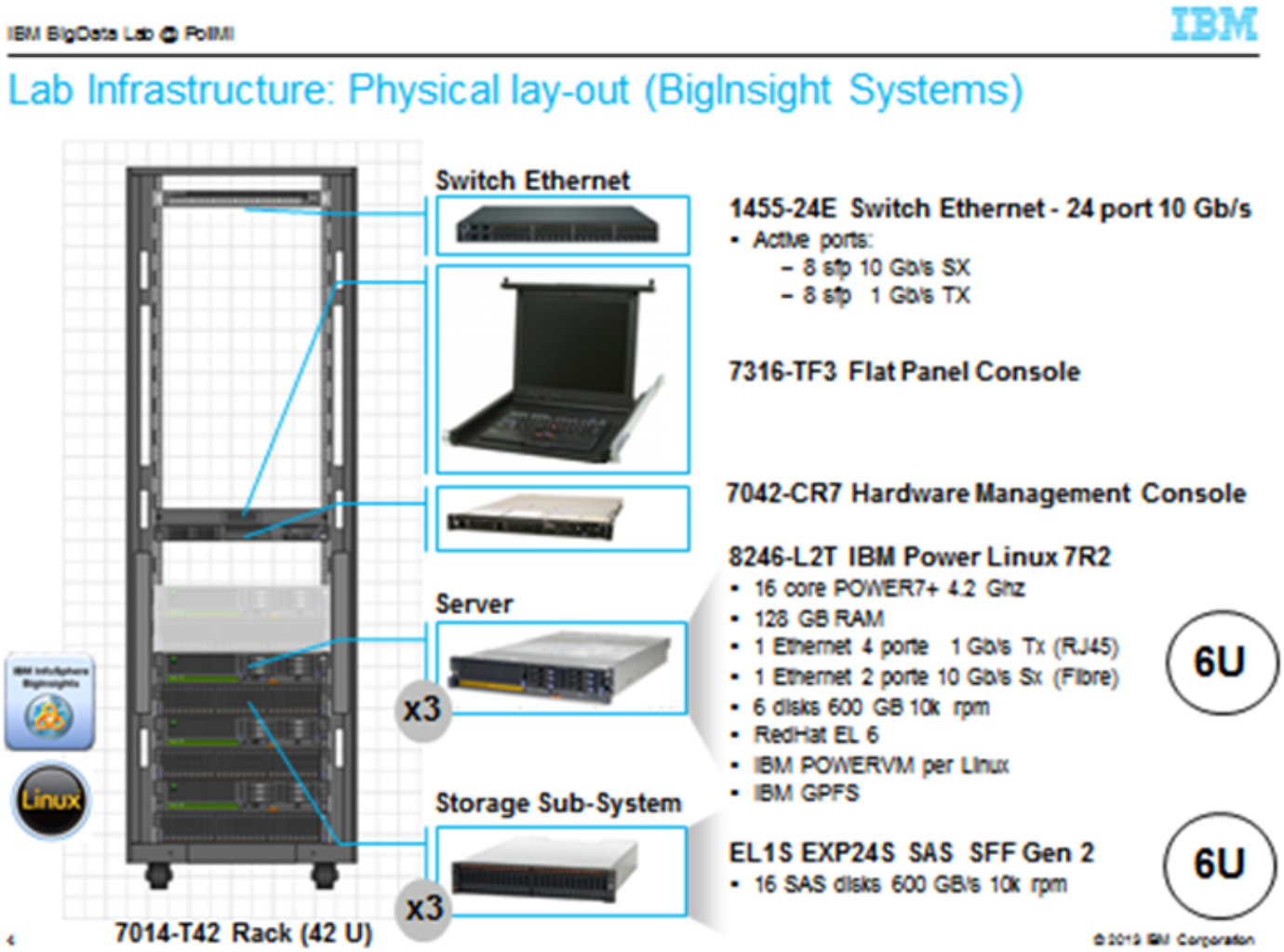
Single DB2 node for "In-memory computing"

Sistema IBM Power 730

- 8 core POWER7+ 4.3 Ghz / 512 GB RAM
- AIX 7.1

Storage System Expansion

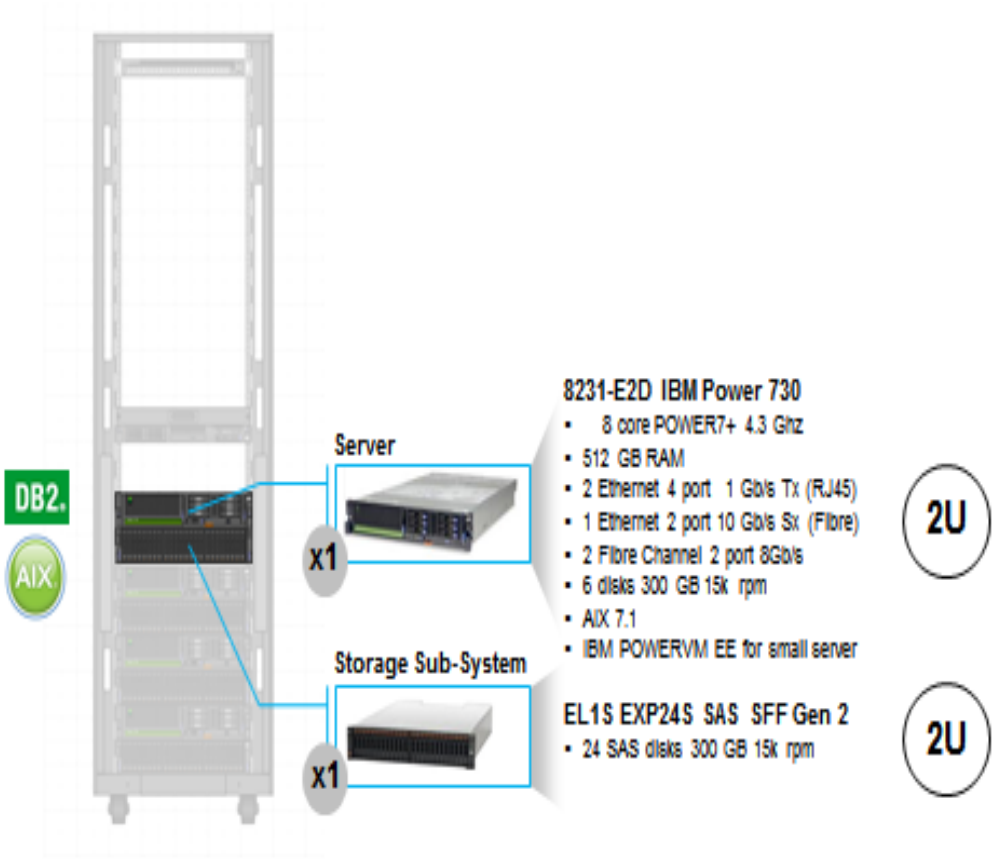
BigData HW Configuration

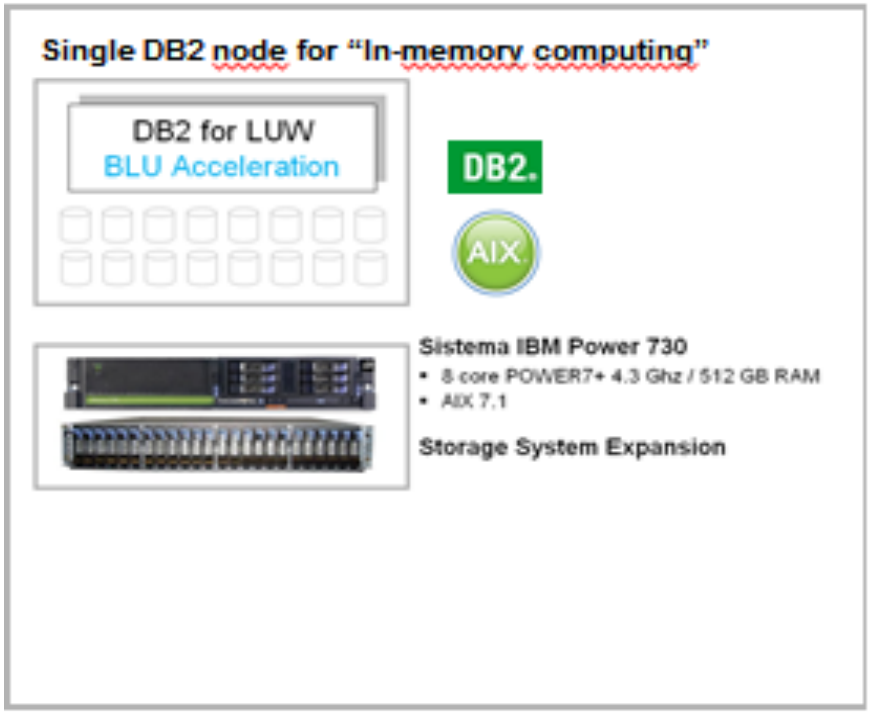


BigData HW Configuration

IBM

Lab Infrastructure: Physical lay-out (DB2 LUW with BLU Acceleration)

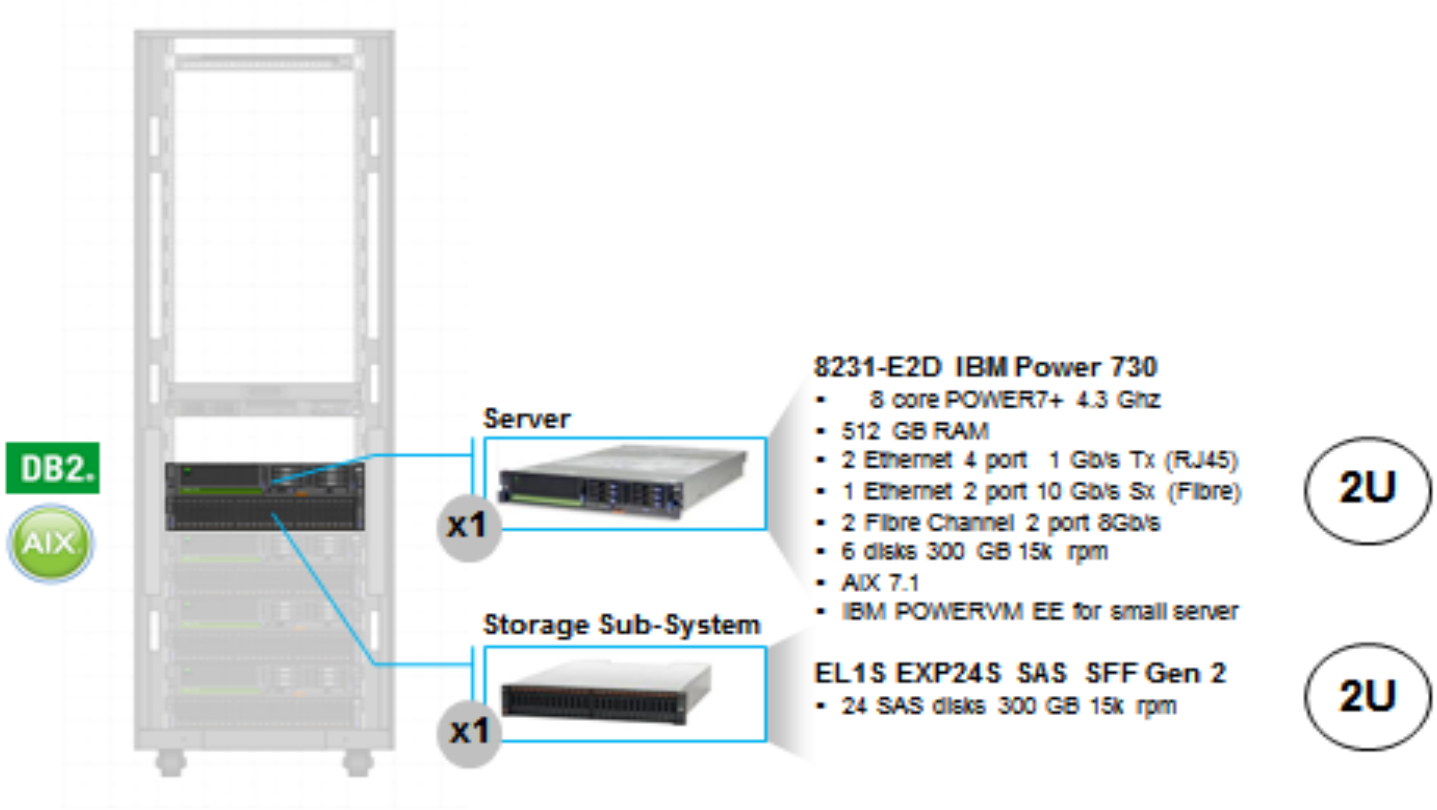




BigData HW Configuration

IBM

Lab Infrastructure: Physical lay-out (DB2 LUW with BLU Acceleration)



BigData SW Configuration

Now:

- IBM Infosphere BigInsights Enterpise Edition 2.1.0.1 (General Parallel File System – File Placement Optimizer (GPFS-FPO), Hadoop, MapReduce) – Linux RedHat 6.5
- Data Integration
- DB2 e Blu Accelator – AIX

Next:

- Cognos / SPSS
- Ilog Cplex

IBM Infosphere BigInsights Enterpise Edition

Table 1. Installation summary for the BigInsights components

Node or Component	Required or Optional	Configured Location	Instances	Type	Note
BigInsights console	Required	Install node	Only one instance	IBM	
BigInsights orchestrator	Required	Any node	Only one instance	IBM	
BigInsights scheduler	Required	All nodes	Library	IBM	
Jaql server	Optional	Any node	Only one instance	IBM	
NameNode	Required	Any node	Only one instance	Hadoop	It is recommended that the NameNode be the only
JobTracker	Required	Any node	Only one instance	Hadoop	It is recommended that the JobTracker be the only
Secondary NameNode	Required	Any node	Only one instance	Hadoop	
DataNode and TaskTracker	Required	Any node	At least one instance	Hadoop	
Derby	Required	Any node	Only one instance	Open source	
Flume	Required	Any node	At least one instance	Open source	
Flume Master	Required	Any Flume node	At least one instance	Open source	
HBase master server	Optional	Any node	Only one instance	Open source	
HBase region server	Optional	Any node	At least one instance	Open source	
Hive	Optional	All nodes	One instance per node	Open source	
Oozie	Optional	Any node	Only one instance	Open source	
ZooKeeper	Required	Any node	At least one instance	Open source	
Avro	Required	All nodes	Library	Open source	
Lucene	Required	All nodes	Library	Open source	
Pig	Optional	All nodes	Library	Open source	
Jaql	Required	All nodes	Library	IBM	
BigInsights text analytics	Required	All nodes	Jaql module	IBM	
R	Required	All nodes	Jaql module	IBM	
Avro Jaql module	Required	All nodes	Jaql module	IBM	
HBase Jaql module	Required	All nodes	Jaql module	IBM	
Lucene Jaql module	Required	All nodes	Jaql module	IBM	

IBM Infosphere BigInsights Enterpise Edition
Key Areas

- **Built-in analytics**
  - **Text analytics**: a vast library of extractors enabling actionable insights from large amounts of native textual data.
  - **Social Data Analytics Accelerato*r***: takes and processes large volumes of social media data, yielding key insights that can be used to develop programs/applications (ex. customer retention and acquisition, campaign effectiveness).
  - **Machine Data Analytics Accelerator**: provides the capability to ingest and process large volumes of machine data sources, including IT machines, sensors, meters, GPS devices and more.
  - **Big R**: Big R enables the use of R as a query language to explore, visualize, transform, and model big data right from their R environment and without any explicit programming using MapReduce or Jaql.

IBM Infosphere BigInsights Enterpise Edition
Key Areas

- **Usability**
  - **Big SQL**: SQL on Hadoop. It provides a single point of access and view across all big data.
  - **BigSheets**: Web-based analysis and visualization tool with a spreadsheet-like interface, featuring D3 graphs, enabling large amounts of data analysis and supporting design and management of data collection jobs.
  - **Development Tools**: Eclipse based development environment for building and deploying analytic applications.
  - **Management Console**: Auditing helps tighten security and access control, while monitoring provides the ability to control all applications from a centralized dashboard.
  - **GPFS-FPO**: Provides POSIX compliant, enterprise-class big data distributed file system capabilities to the Hadoop and MapReduce environment.

IBM Infosphere BigInsights Enterpise Edition
Key Areas

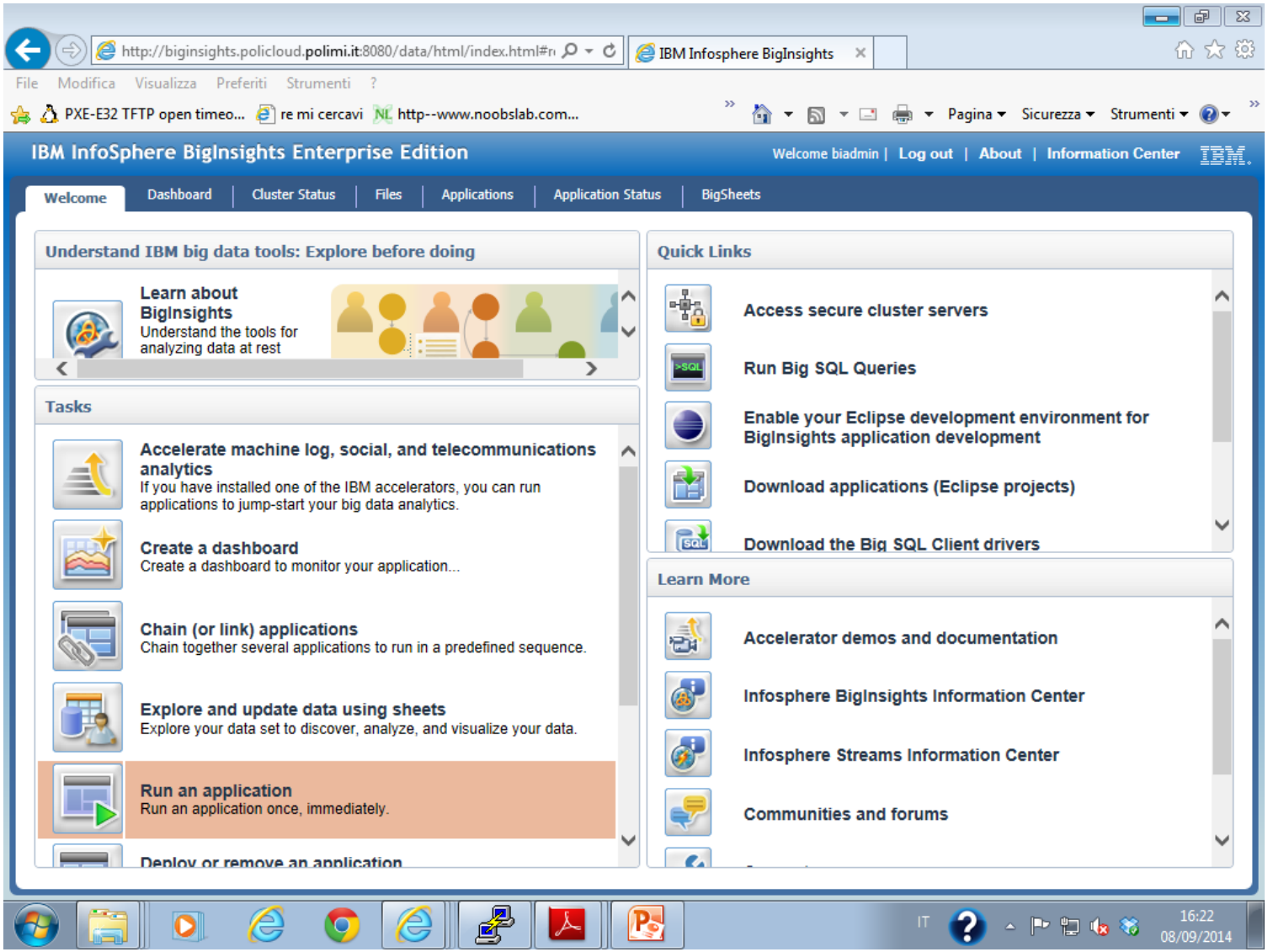
- **Usability**
  - **Workload Optimization**: Adaptive MapReduce adapts to user needs and system workloads automatically to improve performance and simplify job tuning, while workload scheduler provides optimization and control of job scheduling based on user-selected metrics.
- **Enterprise integration**
  - **Big Data Integration**: Codeless creation of data integration logic and jobs, reusable across the enterprise through ETL jobs powered by Information Server. Enable data governance including data lineage, business rule and policy management, data quality.
  - **Data Privacy for Hadoop**: Mitigate risk with sensitive data discovery. Maintain an acceptable risk tolerance with data monitoring, within source systems and on Hadoop itself.

IBM Infosphere BigInsights Enterpise Edition
Key Areas

- **Enterprise performance and integration**
  - **Big Data Integration**: Codeless creation of data integration logic and jobs, reusable across the enterprise through ETL jobs powered by Information Server. Enable data governance including data lineage, business rule and policy management, data quality.
  - **Customer Identification in Hadoop**: Enhance customer analytics by establishing a unique identifier for customer information stored in Hadoop with Master Data Management probabilistic matching.
  - **Data Privacy for Hadoop**: Mitigate risk with sensitive data discovery. Maintain an acceptable risk tolerance with data monitoring, within source systems and on Hadoop itself.

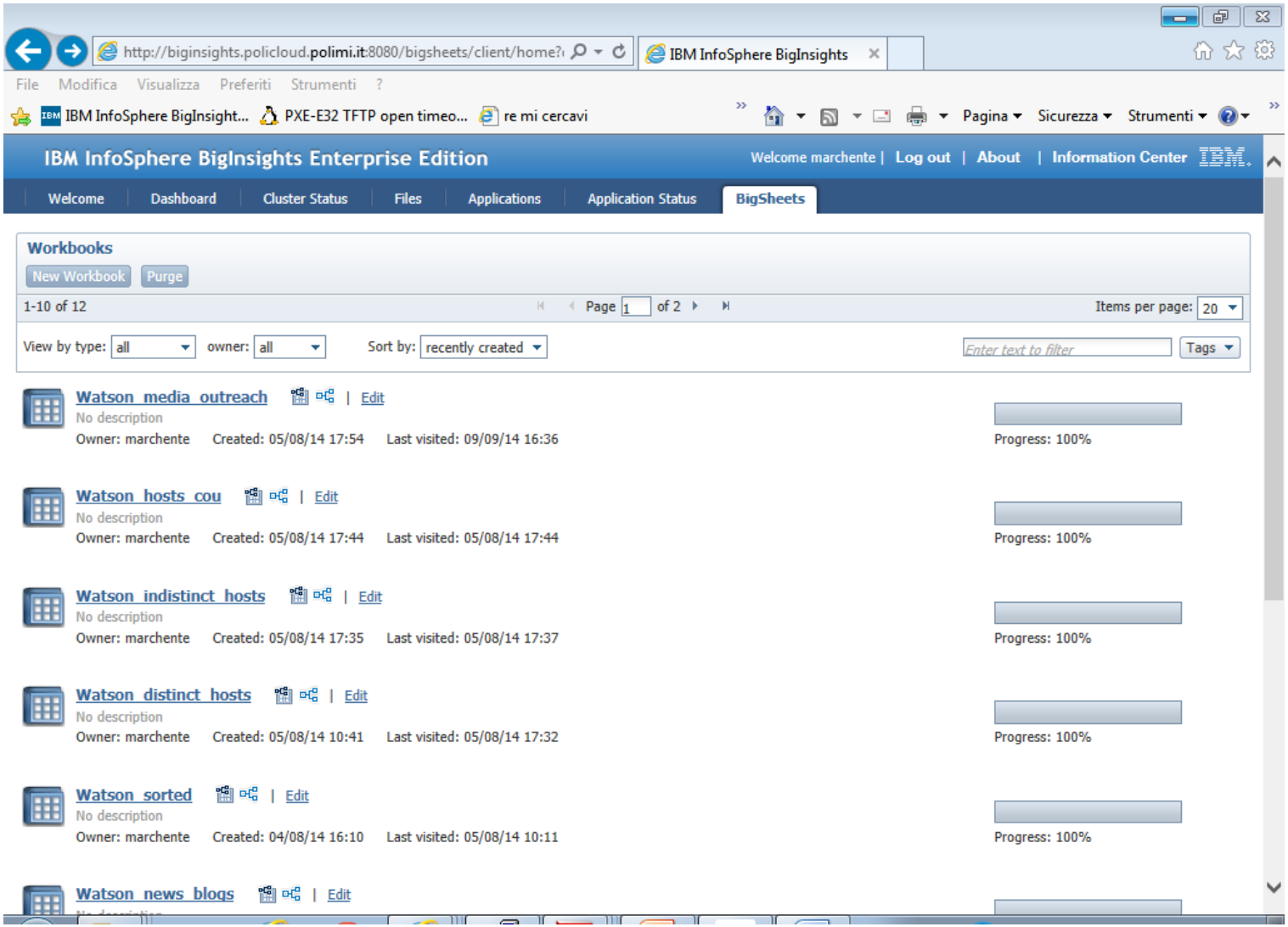
IBM Infosphere BigInsights Enterpise Edition

The Web Interface



IBM Infosphere BigInsights Enterpise Edition

BigSheet Interface



IBM Infosphere BigInsights Enterpise Edition

BigSQL

IBM's SQL interface to its Hadoop-based platform, that:

- provides developers with an environment for querying data managed by Hadoop.
- enables data administrators to create new tables for data stored in Hive, HBase, or their BigInsights distributed file system and to populate Big SQL tables with data from various data sources.
- offers JDBC and ODBC drivers allowing many existing tools to use it for queries.
- is not a replacement for relational DBMS technology (ex. Big SQL tables may contain columns of complex data types such as struct and array, rather than simply "flat" rows).

IBM Infosphere BigInsights Enterpise Edition

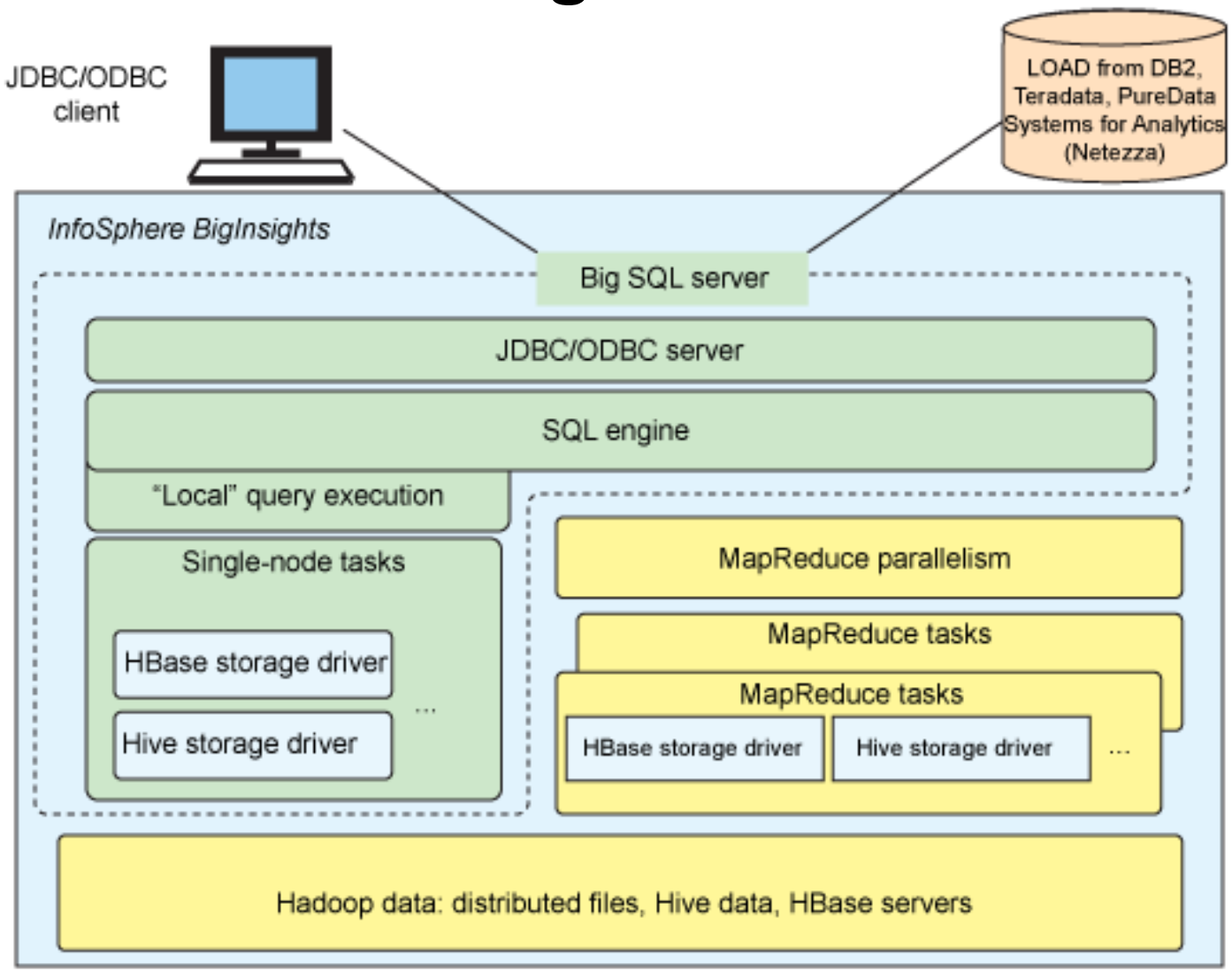
BigSQL

BigSQL supports several underlying storage mechanisms as:

- Delimited files (such as comma-separated files) stored in HDFS or GPFS-FPO
- Hive tables in sequence file format, Record Columnar File (RCFile) format, ..(Hive is Hadoop's data warehouse implementation)
- HBase tables (HBase is Hadoop's key-value or column-based data store)

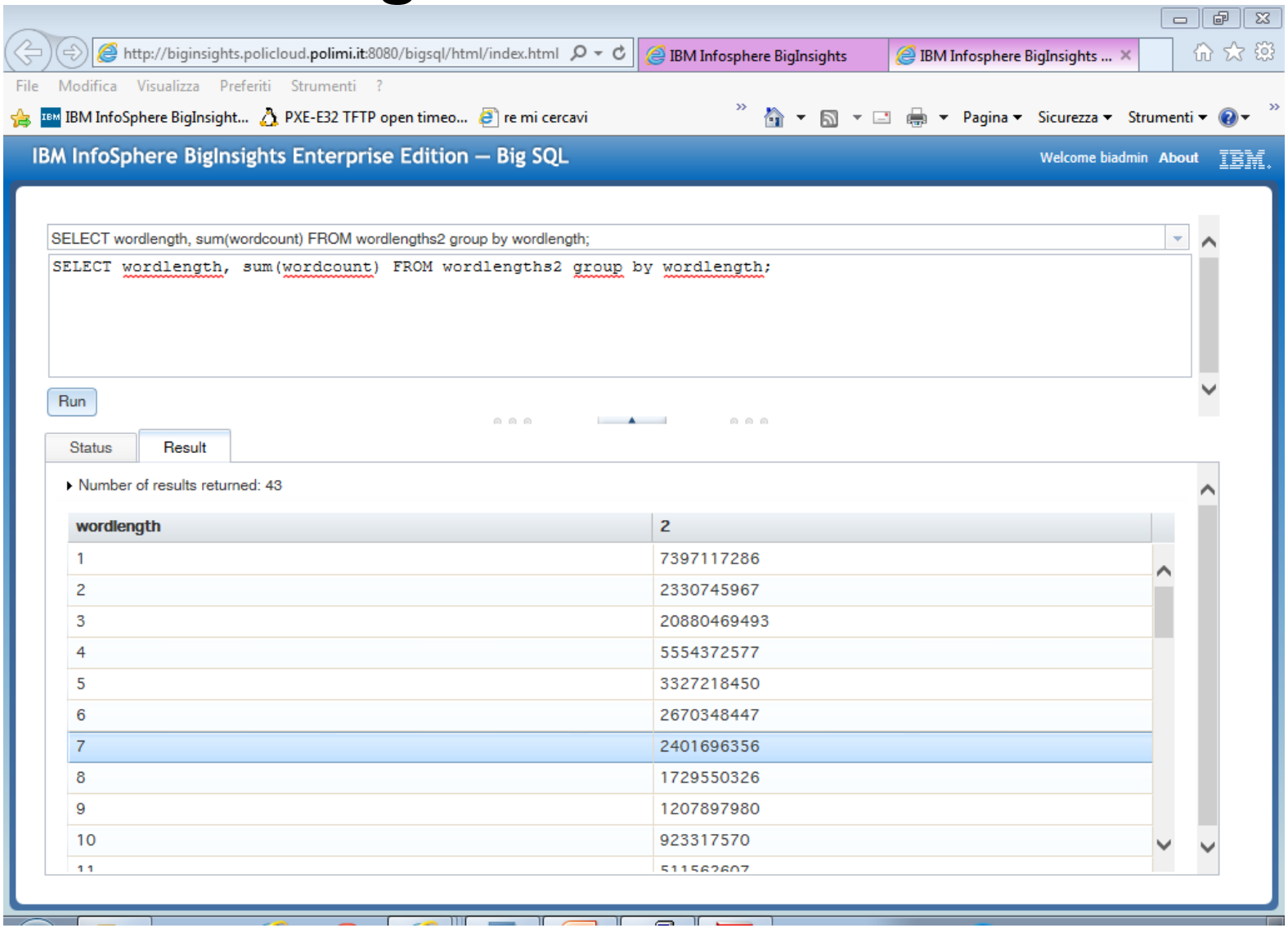
IBM Infosphere BigInsights Enterpise Edition

BigSQL



IBM Infosphere BigInsights Enterpise Edition

BigSQL Web Interface



IBM Infosphere BigInsights Enterpise Edition
BigSQL

A table example

```
DROP TABLE IF EXISTS lineitem;
CREATE TABLE lineitem (
L_ORDERKEY INT, L_PARTKEY INT, L_SUPPKEY INT, L_LINENUMBER INT, L_QUANTITY DOUBLE, L_EXTENDEDPRICE DOUBLE, L_DISCOUNT DOUBLE,L_TAX DOUBLE, L_RETURNFLAG STRING, L_LINESTATUS STRING, L_SHIPDATE STRING, L_COMMITDATE STRING, L_RECEIPTDATE STRING, L_SHIPINSTRUCT STRING, L_SHIPMODE STRING, L_COMMENT STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
STORED AS TEXTFILE;
```

IBM Infosphere BigInsights Enterpise Edition

Apache HBase

A column-oriented not relational database management system(*) modeled after Google's Bigtable (a distributed storage system for structured data designed to scale to a very large size: petabytes of data across thousands of commodity servers) on top of Hadoop and HDFS:

- is well suited for sparse data sets, which are common in many big data use cases.
- does not support a structured query language like SQL;
- adds transactional capabilities to Hadoop, allowing users to conduct updates, inserts and deletes
- applications are written in Java much like a typical MapReduce application.
- may be accessed through the Java API but also through REST, Avro or Thrift gateway APIs

IBM Infosphere BigInsights Enterpise Edition
Apache HBase

HBase:

- is more a "Data Store" than a "Data Base" (it lacks many of the features you find in an RDBMS, such as typed columns, secondary indexes, triggers, and advanced query languages).
- has many features which supports both linear and modular scaling (clusters expand by adding Region Servers that are hosted on commodity class servers).
- his Master node manages the cluster and Region Servers (like HDFS (NameNode and slave nodes) and MapReduce (JobTracker and TaskTracker slaves) operative configurations).

IBM Infosphere BigInsights Enterpise Edition
Apache HBase

Row Key	Time Stamp	ColumnFamily contents	ColumnFamily anchor	ColumnFamily people
"com.cnn.www"	t9		anchor:cnnsi.com = "CNN"	
"com.cnn.www"	t8		anchor:my.look.ca = "CNN.com"	
"com.cnn.www"	t6	contents:html = "<html>..."		
"com.cnn.www"	t5	contents:html = "<html>..."		
"com.cnn.www"	t3	contents:html = "<html>..."		
"com.example.www"	t5	contents:html = "<html>..."		people:author = "John Doe"

```
{
        "com.cnn.www": {
                contents: {
                        t6: contents:html: "<html>..."
                        t5: contents:html: "<html>..."
                        t3: contents:html: "<html>..."
                }
                anchor: {
                        t9: anchor:cnnsi.com = "CNN"
                        t8: anchor:my.look.ca = "CNN.com"
                }
                people: {}
        }
        "com.example.www": {
                contents: {
                        t5: contents:html: "<html>..."
                }
                anchor: {}
                people: {
                        t5: people:author: "John Doe"
                }
        }
}
```

Consider an HBase table more as a multi-dimensional map, than a traditional table with rows and columns.

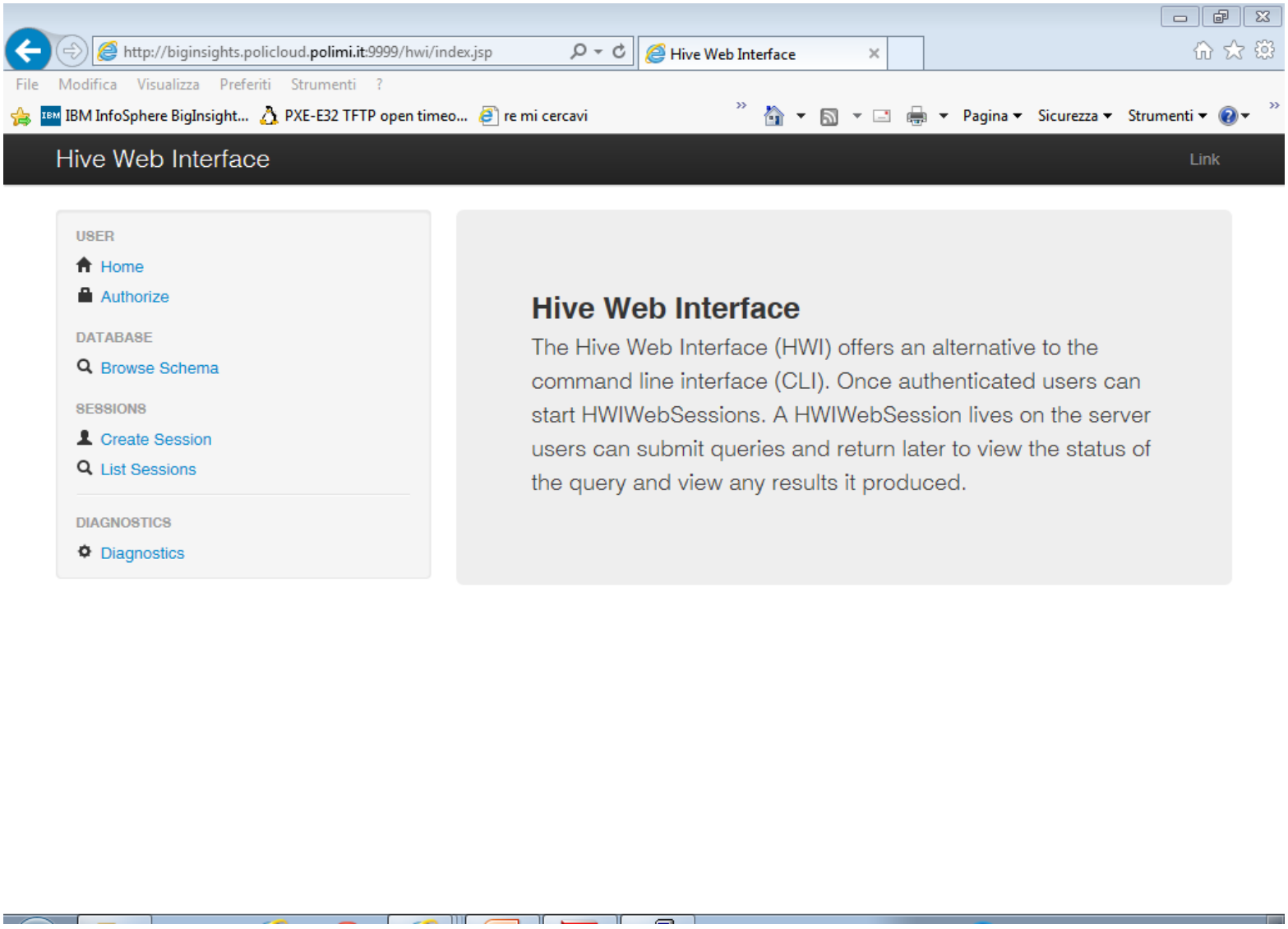
IBM Infosphere BigInsights Enterpise Edition
Apache Hive

A data warehouse infrastructure (*) built on top of Hadoop, to work with large datasets residing in distributed storage. It provides:

- tools that facilitate data Extraction, Transformation and Load (ETL) operation;
- a mechanism to project structure onto this data (stored in the HDFS) and query the data to be analyzed using a SQL-like language called HiveQL
- a way to plug in traditional map/reduce programs when logic expressed in HiveQL is considered not convenient about performances.

IBM Infosphere BigInsights Enterpise Edition

Apache Hive



IBM Infosphere BigInsights Enterpise Edition
Apache Hive

```
CREATE TABLE wordlist2 (word STRING, year INT, wordcount INT, pagecount INT, bookcount INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH '/bigdata/user/biadmin/pighivejaql/Googl.csv' OVERWRITE INTO TABLE wordlist2;

CREATE TABLE wordlengths2 (wordlength INT, wordcount INT);

INSERT OVERWRITE TABLE wordlengths2 SELECT length(word), wordcount FROM wordlist2;

SELECT wordlength, sum(wordcount) FROM wordlengths2 group by wordlength;
```

IBM Infosphere BigInsights Enterpise Edition
Apache Pig

A high-level Hadoop programming language for data analysis programs that provides:

- a data-flow language
- an execution framework for parallel computation

Pig programs structure is suited for substantial parallelization, enabling them to handle very large data sets.

Actually, Pig's infrastructure layer is based on a compiler that produces sequences of Map-Reduce programs, for which large-scale parallel implementations already exists.

IBM Infosphere BigInsights Enterpise Edition

Apache Pig

Pig's language layer currently consists of a textual language called Pig Latin (*), characterized by the following features:

- Ease of programming parallel data analysis tasks.
- Optimization opportunities related to specific task encoding.
- Extensibility through functions developed for special-purpose processing.

Pig supports running scripts (and Jar files) that are stored in HDFS, Amazon S3, and other distributed file systems.

IBM Infosphere BigInsights Enterpise Edition

Apache Pig

Pig runs (execute Pig Latin statements and Pig commands) using various modes.

a) Interactive Mode (both Local/ Mapreduce Mode) via a shell called grunt invoked by command

```
$pig ….
```

or

```
$java -cp pig.jar org.apache.pig.Main …
```

b) Batch Mode (both Local/ Mapreduce Mode) calling a pig script (file.pig) with command

```
$pig file.pig …
```

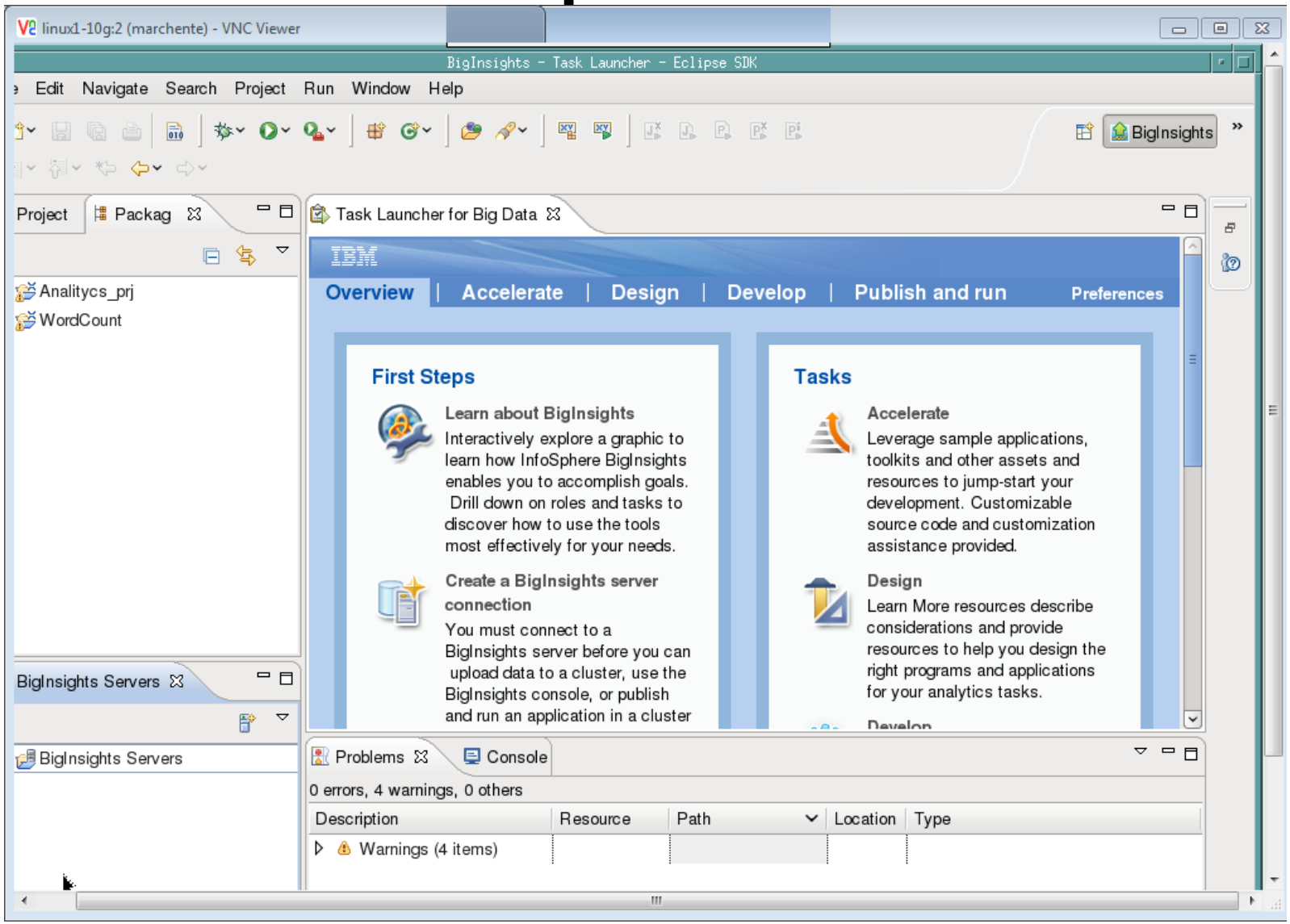
IBM Infosphere BigInsights Enterpise Edition

Apache Pig

```
REGISTER /opt/ibm/biginsights/pig/contrib/piggybank/java/piggybank.jar;

records = LOAD 'pighivejaql/Google.csv' AS (word:chararray, year:int,
wordcount:int, pagecount:int, bookcount:int);

grouped = GROUP records by
org.apache.pig.piggybank.evaluation.string.LENGTH(word);

final = FOREACH grouped GENERATE group, SUM(records.wordcount);
DUMP final;
```

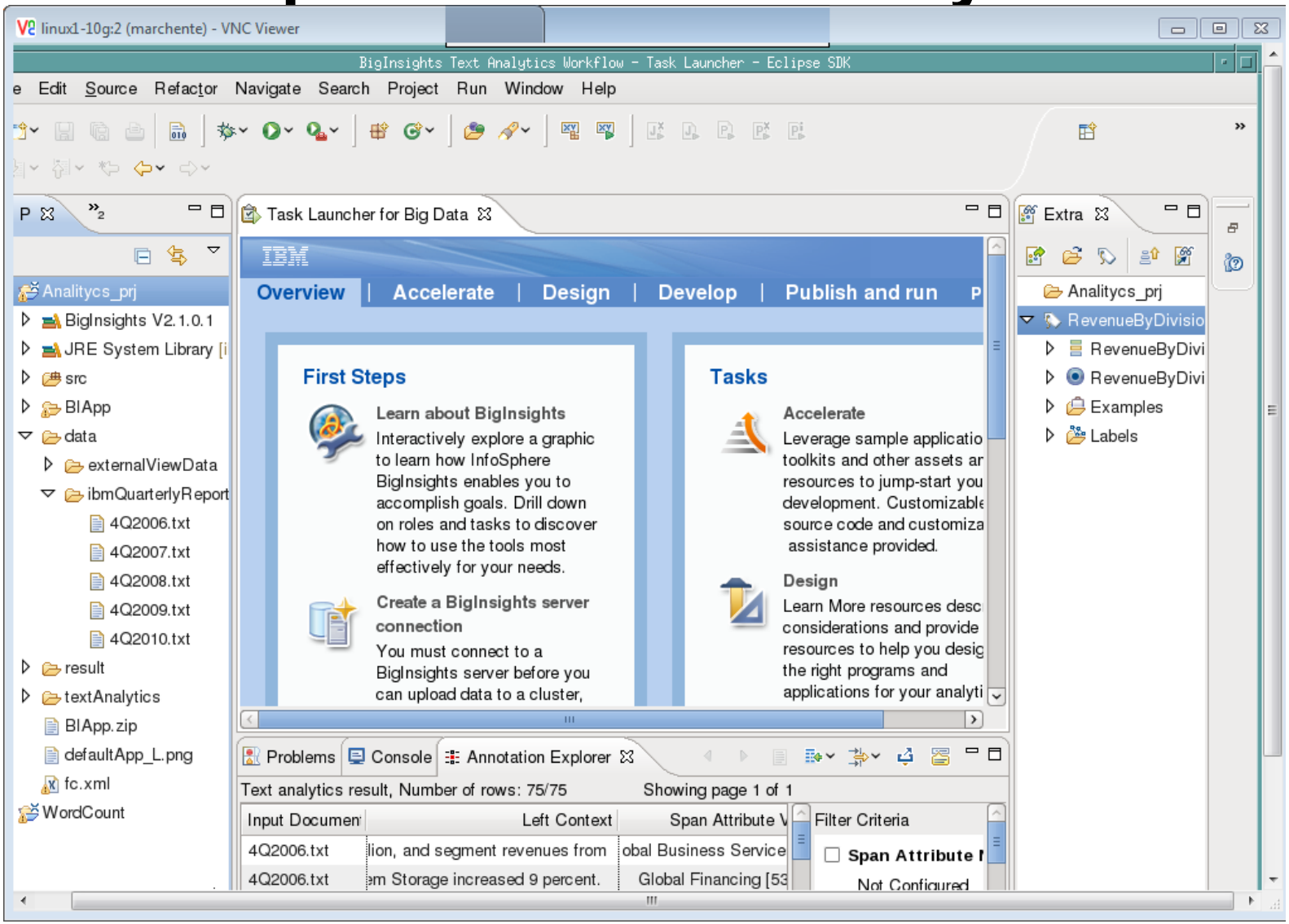
IBM Infosphere BigInsights Enterpise Edition

Eclipse SDK

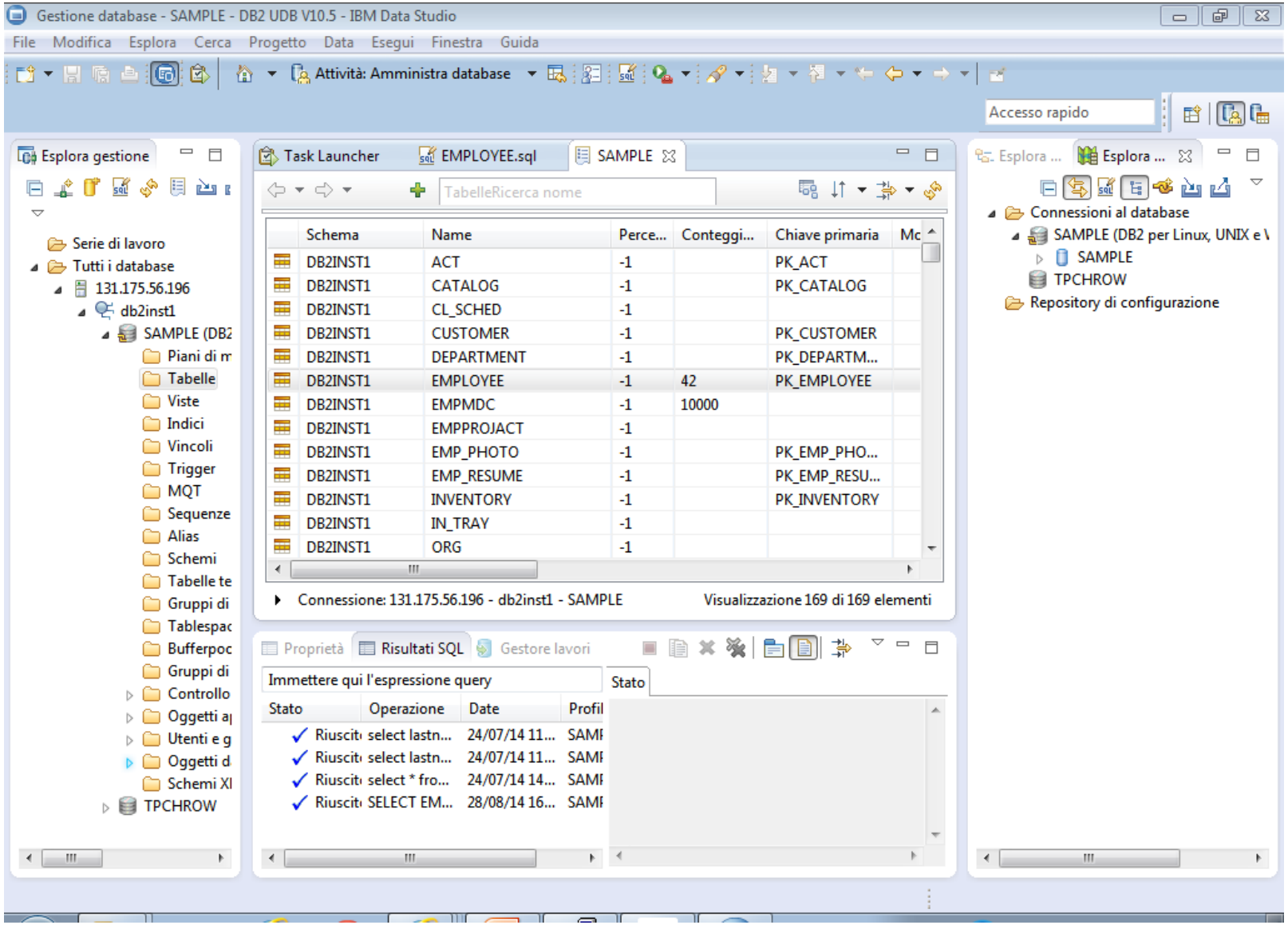


IBM Infosphere BigInsights Enterpise Edition

Eclipse SDK – Text Analytics



DataStudio – IBM DB2 Visual Client



DataStudio – IBM DB2 Visual Client